Attorney's Docket No.: 10699-003003

APPLICATION

FOR

UNITED STATES LETTERS PATENT

TITLE: TRANSFER OF MESSAGES IN A MULTIPLEXED SYSTEM

APPLICANT: ANTONIO CANTONI AND ROBERT M. NEWMAN

CERTIFICATE OF MAILING BY EXPRESS MAIL

Express Mail Label No. EL224699579US

I hereby certify under 37 CFR §1.10 that this correspondence is being deposited with the United States Postal Service as Express Mail Post Office to Addressee with sufficient postage on the date indicated below and is addressed to the Commissioner for Patents, Washington, D.C. 20231.

July 31, 2001
Date of Deposit

Samantha Bell
Signature

Samantha Bell
Typed or Printed Name of Person Signing Certificate

TRANSFER OF MESSAGES IN A MULTIPLEXED SYSTEM

This invention relates to a transfer of messages in a multiplexed system.

More particularly, the invention relates to a method for the transfer of messages in a time multiplexed slotted environment such as a communications network. The network can be of the type disclosed in International Publication No. WO 86/03639 and that disclosure is incorporated herein by cross-reference.

Generally speaking, the invention provides an efficient method for the connectionless or connection oriented transfer of messages of arbitrary but finite length in a time multiplexed slotted environment with constrained destination resources.

The method provides for the efficient support of any type of addressing (short or extended, hierarchical or non-hierarchical) in the one environment, even in a system with short slots.

In the one environment, the method can guarantee delivery of messages or provide a more efficient transfer at the expense of occasional message loss. In either case, there are minimal communication overheads and the utilisation of destination resources is maximised. Thus the method provides a wide range of options and considerable scope for achieving a range of performance-cost objectives.

BACKGROUND ART

In the data communications environment, information is generally exchanged in units called packets. These consist of an overhead necessary for the control and addressing of the unit through the data switch and of the actual information. Typically the size of the information unit is not fixed but depends upon the message and the amount of information to be transferred.

Early packet switches handled the variable length packets as a whole unit, allocating all its communication resource to the transfer of the packet until its completion. There are a number of new packet switch designs emerging that switch only small fixed length slots. These switches are commonly referred to as Fast Packet Switches. Such switches are an improvement since they are generally simpler, may operate at higher speeds and allow for the support of real-time traffic.

If the Fast Packet Switch is to carry packet communications of variable length then it is necessary that the original packet message be segmented for transmission over the switch and reassembled at the destination. The segmentation function is relatively simple only requiring that the message be divided into units of size equal to or smaller than the slot size. The transmission of the segments over the packet switch however requires much more since it is necessary that the destination can receive and order all of the segments of the message. Thus it is required that there be a logical association between all of the slots of a single message. The reassembly function then reconstructs the original variable length message from all of the received segments. Special care may need to be taken in the reassembly function to allow for the possibility that more than one message may need to be reassembled concurrently.

Some schemes have been developed to perform the segmentation and reassembly function. However, these are limited either in the efficiency of the transport in the switch or in the performance of the reassembly func-

tion. The efficiency of the transport is typically limited by the overheads that are carried on each slot. These overheads are required to route the slot to the destination and to control the reassembly of the message at the destination. An example of this is the Slotted Ring protocol where 13 bytes of overhead are required in each slot as described in I E.E.E. 802.6 Draft Standard Proposal "Slotted Ring" Sept. 1986 The overheads in that case include addressing, sequence indication, and length indication.

The problem with the slot overhead is compounded by addressing requirements. The common address field sizes used in data communications are 16 to 48 bits. With 48 bit addressing there is an overhead of 12 bytes per slot (source and destination address) in addition to the reassembly overheads. This approach to segmentation is clearly inefficient with small (less than 32 bytes) slot sizes.

The overhead problem can be reduced by logically associating the segments of the same message by the use of a count scheme, as suggested in an article by K. Yukimatsu, N. Watanabe, T. Honda "Multicast Communication Facilities in a High Speed Packet Switching Network", Proc. ICCC 86 Munich Sept. 1986. pp 276–281. In this approach the segments of the message are transmitted with a two octet i.e. 16 bits overhead, a count field. The count field gives the number of slots separation between two consecutive segments of the same message. By the use of this count the destination can determine all slots of the message. The limitation with this approach is that the number of slots between consecutive segments of the message is limited by the maximum value of the count field. Also, in the case of a multiple access switch, the source cannot transmit more than one message at a time. This reduces the efficiency of transfer when connection orientated reassembly schemes are used.

BRIEF DESCRIPTION OF THE INVENTION

According to the present invention there is provided a method of transmitting variable length messages on a network from a source to a destination in fixed length slots which include a header field and a message segment, said method including the steps of providing a source identifier field in the header field of each slot, said source identifier field including a source identifier code which is uniquely associated with the message to be transmitted, transmitting the slots on the network, and controlling the reassembly of slots at the destination in accordance with the source identifier codes of the slots received at the destination.

The invention also provides an apparatus for transmitting variable length messages on a network from a source to a destination in fixed length slots said apparatus including:

a segmentation machine for segmenting the message into fixed length slots which include a header field and a message segment, said machine including coding means for providing a source identifier field in the header of each slot, said source identifier field including a source identifier code which is uniquely associated with the message to be transmitted, and a reassembly machine located, in use, at the destination, said reassembly machine including control means for controlling reassembly of slots in accordance with the source identifier codes of the slots.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. 1 is a diagrammatic representation showing a variable length message and fixed length segments;

FIG. 2 is a block diagram showing the segmentation and reassembly machines coupled to a fast packet switch;

FIG. 3 diagrammatically illustrates the segmentation and reassembly machines coupled to a QPSX network;

FIG. 4 is a block diagram of an access unit (AU) of the network shown in FIG. 3;

FIG. 5 is a more detailed block diagram of the access unit;

FIG. 6 is a segmentation state machine diagram;

FIG. 7 is a diagram showing request and confirm primatives for segment transfer to the access unit;

FIG. 8 is a reassembly state machine diagram;

FIG. 9 is a diagram showing an indication primative for segment transfer from an access unit;

FIG. 10 is another reassembly state machine diagram;

FIG 11 is a block diagram of a reassembly machine; and FIGS. 12A and 12B are flowcharts illustrating logical operations in the reassembly machine.

DETAILED DESCRIPTION OF THE EMBODIMENTS OF THE INVENTION

FIG. 1 diagrammatically shows a message 20 which is of variable length. The message includes address fields 22 and 24 for the destination address (DA) and source address (SA). The message includes a length field (L) 26 and an information field (IU) 28 followed by an error checking field (CRC) 30. The error checking field 30 can be of any known type.

In accordance with the invention, the variable length message 20 is segmented into a number of slots 32 of equal length. The method places the address field 22 and 24 in the first of the slots 32 and subsequently logically associates the following slots of the message with the first slot using a unique identifier, as is diagrammatically illustrated in FIG. 1.

Each slot 32 has an Access Control Field (ACF) 34, TYPE field 36 and Source Identifier (SI) field 38. In accordance with the invention, the SI field 38 is used to provide the logical linking between slots 32 of the same message. The slots 32 also include an information field 40 which is used to carry information from the original message 20. The ACF 34 is related to controlling transmission of the slots through the network.

The TYPE field 36 within the slot can be used for the indication of a wide range of information. In one use it can allow for the separation of different communication classes within the one environment. As an example the messages with different field sizes and structures could be carried on the same switch. Also the TYPE field can be used to identify between different slot structures. Other segmentation schemes such as those described in the Background Art could be used in the same environment and distinguished by the TYPE field.

The TYPE field specifies how the SI field 38 and the information field of each slot should be interpreted depending on, among others, the following factors:

(a) Whether connectionless or connection oriented communication is supported.

(b) Whether short or long source and destination addresses are to be supported.

(c) Whether extended hierarchical or non hierarchical source and destination addressing is supported.

(d) Whether the slot is carrying the first, last or continuation segment of a message.

(e) Whether the message fits in a single segment.

(f) Whether or not the source supports message transmission to multiple destinations concurrently.

The SI field 38 is a label which enables the logical association of all segments 40 belonging to the one message and hence enables them to be reassembled into the original message 20.

FIG. 2 diagrammatically illustrates a communications system comprising a source 42 which produces messages 20 of variable length for transmission on a network or fast packet switch 44 to a destination 46. The system includes a segmentation machine 48 coupled between the source 42 and the switch 44 and a reassembly machine 50 coupled between the switch 44 and the destination 46. The segmentation machine 48 converts messages 20 of indefinite length to slots 32 of fixed length for transmission on the switch 44. The reassembly machine 50 reassambles the slots 32 into the original message 20 for input to the destination 46. The segmentation and reassembly machines 48 and 50 would be located at respective nodes or access units coupled to the network.

FIG. 3 illustrates in more detail the use of the segmentation and reassembly machines 48 and 50 in a QPSX network of the type disclosed in WO 86/03639. The QPSX network comprises two unidirectional buses, bus A and bus B with data flowing in opposite directions, a central controller 2 and a number of distributed nodes or access units (AU's) 4 coupled between the buses A and B. Although each bus originates and terminates at the central controller 2 neither has a through connection, in normal circumstances. Each AU 4 has read taps 6 and 8 from the respective buses and lines 10 and 12 connected to unidirectional write couplers to the respective buses. The write transmit only in the direction of propagation of the respective buses. The read connections for each AU are attached to the bus ahead of the write connections and consequently the information read by each AU is uneffected by that written by it. In the illustrated arrangement, a source 42 is coupled to one of the access units 4 via the segmentation machine 48. The access unit transmits the message in fixed length slots on the network to the access unit 4 associated with the destination 46. Normally each access unit would have both segmentation and reassembly machines 48 and 50 to enable two way communications. The segmentation and reassembly machines 48 and 50 can be regarded as part of the interface IP 16.

FIG. 5 shows in more detail the connection of the segmentation and reassembly machines 48 and 50 to an access unit 4 of the type described in the aforementioned publication. FIG. 5 corresponds generally to FIG. 12 of that specification and hence need not be described in more detail here.

Each node in the network will have one or more unique SI's. Each SI can be used by the node for the transfer of a message. When the message transmission is complete the SI can be reused. Multiple SI's for a single node allow that node to transfer more than one message concurrently

To describe the operation of the message transfer scheme, the segmentation of the message into slots is considered first and the action at the receiver is considered after that.

The train of slots 32 sent by the segmentation machine 48 is shown in FIG 1 The first slot of a multiseg-

ment message will be identified as such by a BOM (Beginning Of Message) code in the TYPE field 36. The SI field 36 is set by reference to the unique SI of the source node and the information field 40 contains the first segment of the message. Thus the DA field 22 of the message 20 is at the head of the information field 40. The following segments of the message until the last are each placed in the information fields of slots with the TYPE field 36 set to COM (Continuation Of Message) and the SI field 38 containing the source's code which is unique for this message. The last slot of a multisegment message has the TYPE field 36 coded as EOM (End Of Message), as shown.

For the transfer of a message 20 that only requires a single slot 32 the SSM (Single Slot Message) code is used in the TYPE field 36. The SI is not required in this case, however it is still used for consistency in operation.

An implementation of the segmentation machine 48 will now be described with reference to the state diagram which is shown in FIG. 6. In this diagram, the condition for a state transisition is shown above the transition line and the action taken is below the line, in accordance with standard notation. This state machine will handle the receipt of a message from a single source at a time. If simultaneous receipt of more than one message is required, multiple state machines would be required.

To specify the commuunications between adjacent parts of the system, there are three communications primitives, as follows:

(i) Request (REQ): This is a request to send a unit of data,

(ii) Indication (IN): This is an indication that a unit of data has been received, and

(iii) Confirmation (CONFIRM): This is a confirmation that a unit of data has been sent without error.

Between the machines 48 and 50 and the fast packet switch 44, (via the access units 4) communication primitives are prefixed by SAR_ACC hence there are three communication primitives as follows:

SAR_ACC Request

SAR_ACC Indication

SAR_ACC Confirm.

Also for source or destination equipment 42 and 46 such as a computer attached to the segmentation and reassembly machines 48 and 50, communication primitives are prefixed by APP_SAR hence there are three communication primitives as follows:

APP_{13} SAR Request:

APP_SAR Indication

APP_SAR Confirmation

A segmentation machine operates on an unconfirmed APP SAR request, provided that sufficient resources are available. Tags for SAP_ACC requests and Source Identifiers (SIs) are common resources for all of these machines. Tags used in SAP_ACC requests are unique over all such requests from any segmentation machines. A tag is allocated on an SAR_ACC request and deallocated on an SAR_ACC confirm. The tag for the segments are local to the particular segment at the segmentation machine and is not transmitted on the network. Tags are reused when message transmissions have been completed. The tag is coded in TAG fields 52 and 54 in Request and Confirm primitives 56 and 58. The TAG fields 52 and 54 in the Request and Confirm primitives 56 and 58 are used for communications between the segmentation machine 48 and access unit 4, as

seen in FIG. 7. The Request primative 56 comprises the TAG field 52. T-SEG field 60 together with the slot 32. The fields 60 and 52 are control fields which are not transmitted beyond the access unit 4. The Confirm primative 58 comprises a TAG field 54 and TR field 64, being control fields for communication between the access unit 4 and the segmentation machine 48. The codes in the TAG fields 52 and 54 must correspond for a particular segment 40. Therefore a particular message 20 will have a number of TAG codes corresponding to the various segments 32 required to be used for the transmission. Successful transmission or otherwise is indicated by the coding in the TR field 64.

Source Identifiers (SI) are selected at the source so that the SI will identify a unique message to any reassembly machine. The SI is allocated on an APP_SAR request and is deallocated in an implementation dependent manner.

One reasembly machine 50 is needed at the destination 46 for each SI. A reassembly packet timer (not shown) is also associated with each reassembly machine, to prevent locking in a wait state for signals.

An SAR_ACC confirm is routed to the segmentation machine which generated the SAR_ACC request. An SAR ACC indication is routed to the reassembly machine associated with the SI. The segmentation function of the segmentation machine 50 is described in the following paragraphs.

1.1 ABBREVIATIONS USED IN THE SEGMENTATION STATE DIAGRAM OF FIG. 6

CONFIRMSBACK (TAG)	CONFIRMSBACK is an array of flags indexed by tag number A CONFIRMSBACK flag is set when the SARACC confirm has been received corresponding to the SARACC request with tag number TAG
ACB.	All Confirms Back. Set if CONFIRMSBACK(1) is set for all segments.
CHECKSUM	Flag for checking sum of bits successfully transmitted
LAST SEGMENT SENT	Flag indicating whether all segments for this message has been sent.
LAST SEGMENT SENT	Flag indicating that not all segments for this message has been sent.
SI:	The source identifier from the segment in an SARACC indication.
TR.	The TR field 64 from the SARACC confirm indicates SUCCESS or FAILURE in the transmission of a segment.
TSEG	The TSEG field 60 in the SAR ACC request indicates whether the segment is the first segment of the message (BOM), a continuing segment (COM), or the final segment (EOM)

The states of the state machine diagram are as follows:

1.2 STATE S0:IDLE

State S0 is the initial state for all segmentation state machines. In this state, no APP_SAR request is outstanding from the source 42.

S(00) Idle—When an SAR_ACC confirm is encountered in the idle state for a previously confirmed message, the primitive is discarded with no state change.

S(01) Sending—An APP_SAR request causes the initiation of the machine and the transition to the Sending state. The segmentation machine will remain in the Sending state until the message has been sent and all expected responses have been received. A source identifier is allocated.

1.3 STATE S1: SENDING

In state S1, the segmentation machine transmits segments of the message until all segments have been transmitted.

S(10) Idle—The reception of a SAR_ACC confirm with TR=FAILURE indicates a problem with the tramsission. Sending is aborted and the machine, in transition to the idle state, discards outstanding SAR_ACC confirms. The source 42 is notified with an confirm (FAILURE) and the SI is deallocated.

S(11a) Sending—The segmentation machine forms segments from message data, sets the appropriate T SEG and causes an SAR_ACC request for each segment. The segmentation machine also checks the CHECKSUM for the messgage. If the last segment is sent it constructs the appropriately coded segment then sets flag indicating last segment sent.

S(12) Wait Confirm—All segments have been sent and SAR_ACC requests are outstanding. The segmentation machine waits for confirms on any outstanding SAR_ACC requests.

1.4 STATE S2: WAIT CONFIRM

In this state, the segmentation machine waits for expected SAR_ACC confirms.

S(20a) Idle—All SAR_ACC onfirms with TR=SUCCESS are received. An APP_SAR confirm (SUCCESS) is generated indicating a best effort was made of delivery. The SI is deallocated.

S(20b) Idle—SAR_ACC confirm with TR=FAILURE was received indicating the attempt at delivery was unsuccessful. An APP_SAR confirm (FAILURE) is generated indicating failure of delivery to the source 42. The SI is deallocated.

S(22) Wait Confirm—An SAR_ACC confirm with SUCCESS is recorded in the CONFIRMS_BACK array of flags.

The segmentation machine 50 described in the state diagram of FIG. 6 could be implementated by those skilled in the art using known logic blocks or by programming of micro-computers.

An implementation of the reassembly machine 50 will now be described with reference to the state machine diagram of FIG. 8.

2.1 ABBREVIATIONS USED IN THE REASSEMBLY MACHINE

RPT: Reassambly Packet Timer

SR RESOURCES AVAILABLE: Whether there are sufficient resources in the reassembly machine to allow this reassembly machine to exit the idle state.

CHECKSUM Is a sum of all octects in a received message 20.

CHECKSUM GOOD: Indicates whether the checksum over message data matches with the value for CHECKSUM count at the segmentation machine 48 and transmitted to the re-assembly machine for error checking

LENGTH: The number of octects in the message field in the SAR_ACC indication (BOM).

T_SEG: The T_SEG field in the SAR_ACC Indication indicates whether the slot 32 is the first segment of the message (BOM), a continuing segment (COM) or the final segment (EOM).

FIG. 9 diagrammatically illustrates the format for indication primitives for communications between the access unit 4 and the reassembly machine 50. In this case, the indication primitive 66 comprises a T_SEG field 68 and the slot 32. There is no need for any control communication between the reassemoly 50 and the access unit 4.

2.2 STATE R0: IDLE

State R0 is the initial state for all reassembly machines. In this state, no message is being reassembled.

R(00a) Idle—An SAR_ACC indication presents a segment with SSM set. A single segment message is assembled. The checksum is computed and if no error is indicated, an APP_SAR Indication presents the complete message to the destination 46.

R(00b) Idle—An SAR_ACC Indication presents a segment with BOM set and sufficient resources are not available to allow the reassembly machine to leave the idle state.

R(01) Receiving—An SAR_ACC Indication presents a segment with BOM set and sufficient SAR resources are available to allow the reassembly machine to leave the idle state. Buffer allocated of size in octects equal to LENGTH indication. Resources are allocated and the data from the segment is buffered.

In FIG. 8, Action 1 beneath the transition line (11) includes the steps of storing the received segment 32 in sequence and computing the CHECKSUM over the message 20. Action 2 in transition lines R0(00a) and R1(10b) includes of reassembling the message 20, computing the CHECKSUM over the message 20. If CHECKSUM IN GOOD then APP_SAR_IND, otherwise discard the message.

2.3 STATE R1: RECEIVING

In this state, the reassembly state machine is reassembling a message. The machine will remain in this state until the complete message has been assembled or until the Reassembly Packet Timer PRT has expired or there is a problem with available resources. The RPT ensures that the reassembly machine is not effectively put out of service waiting for a lost EOM.

R(10a) Idle—When the Reassembly Packet Timer expires, the reassambly machine will return to the idle state, discarding the partially assembled message.

R(10b) Idle—When the complete message has been assembled (EOM received) the reassembly machine computes the CHECKSUM. If the CHECKSUM indicates no error, an APP_SAR indication presents the received message to the destination 46. If the CHECKSUM indicates an error, the reassambly machine returns to the Idle state and releases resources.

R(11) Receiving—On each SAR_ACC indication with T_SEG=COM, the reassembly machine will buffer the segment in sequence.

The reassemble state machine diagram shown in FIG. 8 could be implemented in logic or by appropriate software.

FIG. 9 is another example of a state machine diagram for a reassembly machine which is particularly designed for use with the distributed queue system of the QPSX

network disclosed in the aforementioned international publication.

This receive machine has two states: IDLE and WAIT. In the IDLE state the machine is not currently receiving any message. Thus in this state the machine will check for slots 32 with TYPE field 36 equal to BOM or SSM. In the case a BOM code is received the machine will check the DA in the information field. If the message is addressed to the station the machine enters the second state. In the case that SSM is detected in the IDLE state and the DA field matches, the length and information fields are copied and the AU 4 will indicate the higher layers that a message is received. The receive machine remains in the IDLE state after copying the slot. The higher layers refer to the higher layers in the Open System Interconnections (OSI) reference model.

The WAIT state is used to receive the slots 32 following the first of a multisegment message. In this state the slots with TYPE field 36 equal to COM and the SI equal to that copied from the first slot of the message will be received. The information fields of these following slots are concatenated to form the complete message. New messages addressed to the given station in this state are ignored by the state machine. Further receive machines are required if such messages are to be received. When the last slot of the message is detected by the receive state machine, the machine will copy the information field of the slot, indicating to the higher layers that a message is received and return to the IDLE state. This completes the receipt of the message. To guard against the loss of the EOM slot, which would cause the receive state machine to be locked in the WAIT state, the RPT timer is used. This timer is started after each BOM or COM slot is received. If the timer expires before the next slot is received, a failure in the transfer is assumed. The machine will then clear all copied slots and return to the IDLE state.

CONSTRAINED DESTINATION RESOURCES

In all practical implementations of destination facilities, i.e. resources such as buffers and processing capacity will be limited and loss of slots can occur unless a positive control mechanism is introduced to control communication between source and destination.

The method enables a source seeking use of the destination resources for reassembly of the original message to be temporarily held up until the required resources are available. The method implements a variety of access disciplines. Among others, the method supports first come first served queueing for the destination resources described briefly below.

Control of access to destination resources is via a "ticket" handout scheme in which the "ticket number" establishes the position of source request in a distributed queue. The first encapsulated segment of a message sent by a source is considered by the destination as a request for resources. A "go ahead" message is returned to the source if the required resources are available, otherwise a "ticket" which uniquely identifies the sources position in a distributed queue of requests is returned. When resources become available the destination broadcasts the "ticket number" of the next source to be serviced. Under the assumption that resources are usually available, the delay incurred in waiting for the "go-ahead" is avoided by the source continuing to send segments of a message while it awaits the reply from the destination. In the case of a negative reply, that is a "ticket" is re-

ceived, the source aborts transmission of further segments and resets its transmission pointer to the first segment. Thus the source buffers each message until it can be completely sent.

The TYPE field is used to control the generation of new ticket number and thereby enables various resources access priority schemes to be implemented. The TYPE field is also used to indicate whether slots from part of a message whose transfer is guaranteed.

FIG. 11 is a block diagram showing an implementation of a reassembly machine 50. In this example, the machine is designed to simultaneously handle seven messages and can therefore be regarded as a seven-fold implementation of the reassembly machine which is described in the state machine diagram of FIG. 8. The machine is also able to simultaneously receive a single segment message.

The machine comprises a DA Select circuit 70 which receives fixed length slots 32 from an access unit 4. The circuit includes an SI comparator block 72 which also receives the slots 32 from the access unit 4. Output from the comparator block 72 is connected to the input of a buffer selector circuit 74. Outputs from the buffer selector 74 are coupled to buffers 77 of a partitioned buffer space 76. The circuit also includes an SSM buffer 78 which receives output from the DA select circuit 70. Outputs from the buffer 78 and buffers 77 are transferred to the destination 46 on output line 80, in the form of a reassembled variable length message 20.

The main function of the DA select circuit 70 is to check the destination address within the BOM and SSM segments copied from the slots 32 received by the access unit 4. If the destination address matches that of the access unit then the DA select circuit 70 will initiate the receipt and reassembly of the whole message provided resource is available to do so.

The SI (Source Identifier) comparator block 72 comprises seven identical sub-blocks (actually any number of sub-blocks may be used. The number of sub-blocks specifies the number of multiple segment messages 20 that may be reassembled simultaneously). The function of an SI_Comp sub-block 82 is to copy all segments 40 with the same SI code in the header of the slot 32. In this way all segments 40 of a message 20 are received and allows the message to be reassembled. When there are more messages destined to an access unit than there are SI_Comp sub-blocks 82, the extra messages will be lost.

The buffer selector 74 is a simple function that directs where a copied segment should be stored. This block operates under the control of the SI comparator circuit 72. Whenever any sub-block 82 detects that a segment should be received the buffer selector 74 is directed to copy that segment into the associated buffer 77.

The buffers 77 are used to store segments 32 while messages are being reassembled. The SSM buffer 78 is used to store all single segment messages. Single segment messages do not need reassembly since the complete message is contained within the one segment. Hence the segment is copied directly to the SSM buffer 78. From there it may be passed out as a reassembled message 20 to the attached destination 46.

The seven buffers 77 of the buffer space 76 are each uniquely associated with an SI_Comp sub-block 82 and each used to store all segments 40 of a common message 20. Each buffer partition must be of sufficient size so as to receive all segments of a message in order to guarantee that no part of the message is lost. When all seg-

ments of a message are received in a buffer partition then the complete variable length message may be passed out of the buffer. When the message is passed out, the SI_Comp sub-block 82 associated with that buffer partition can be made free and hence used to receive another message.

To further describe the operation of the reassembly machine it is useful to consider the data flow paths through the machine when a slot is passed to the machine from the network. A fixed length slot 32 copied from the access unit 4 will enter both the DA Select circuit 70 and comparator block 72. The DA Select circuit 70 will temporarily store the full slot 32. Each of the sub-blocks 82 will only copy the SI field 38 of the slot.

The DA select circuit 70 will check the type field 36. If the type is COM or EOM then the only action on the segment is to store it temporarily, in case it needs to be transferred to the SI_Comp block 72. If the type is BOM or SSM then the DA Select circuit 70 will check the DA address field 22 within the data segment 32. If the DA address does not match that of the access unit 4, then the segment is discarded. If the DA address does match that of the access unit then the segment is intended for that unit. If the segment is an SSM then the message part of it will be passed directly to the single segment buffer 78. Hence the complete message is received and no further action is required by the reassembly machine except to pass it out as a reassembled message.

If the segment received by the DA select circuit 70 is a BOM with a DA match then the circuit 70 will check if there are any free SI_Comp sub-blocks 82 available to control the receipt of the rest of the segments of the message. If not the segment is discarded. This event occurs when the reassembly machine is already receiving messages on all sub-blocks. This extra message is hence lost.

If there is an SI_Comp sub-block 82 available then the DA select circuit 70 will copy the SI from the segment and load it into the free SI_Comp sub-block 82, hence making that sub-block busy. The segment information field 40, that is the slot 32 minus the segment header fields 34, 36 and 38 is then passed via line 84 to the buffer selector circuit 70. The buffer selector ciruit 76 in turn routes the segment 40 into the buffer 77 associated with the claimed SI Comp sub-block 82. The association is signalled explicity to the buffer selector circuit 74 from the SI_Comp block 82 via line 86.

For COM and EOM segments, that is the segments following the first of a message, the decision to copy these into the buffer is made by the SI_Comp circuit 72. Each sub-block 82 will compare the SI read from the incoming slot via line 88 with its own SI. If no sub-block has a match for the incoming SI then the segment is discarded. If there is a match, the information segment 40 is passed from the temporary storage in the DA select circuit 84 via line 84 to the sub-block 82. As with a BOM segment the buffer selector circuit 74 will route the segment to the buffer partition associated with the SI_Comp sub-block 82 that had a match for the SI. If the segment passed to the buffer partition is an EOM segment the reassembly of that message is complete. Hence the SI_Comp sub-block 82 is made free and the complete message passed out along line 80.

FIGS. 12*a* and *b* are flow charts which illustrate the logical steps involved in the reassemble machine 50, illustrated in FIG. 11.